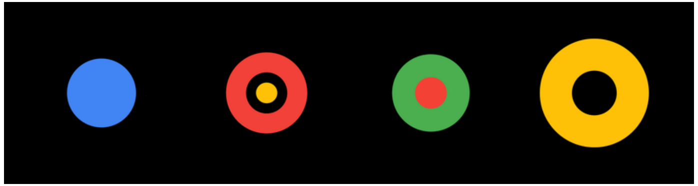
Android UI 进阶

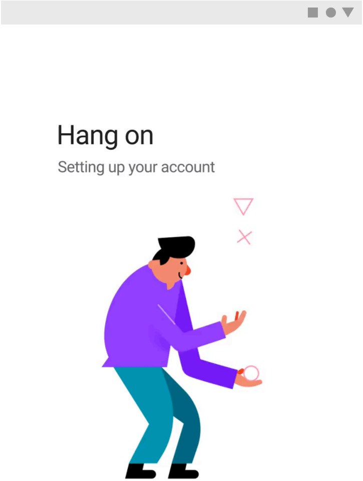
yangjie.bella@bytedance.com - 字节跳动 Android 工程师

Animation

ByteDance字节跳动

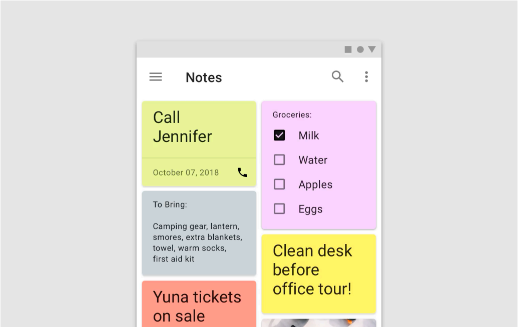
About Animation





Hang on

Setting up your account



Notes

Call Jennifer

October 07, 2018

Groceries:

☑ Milk

□ Water

□ Apples

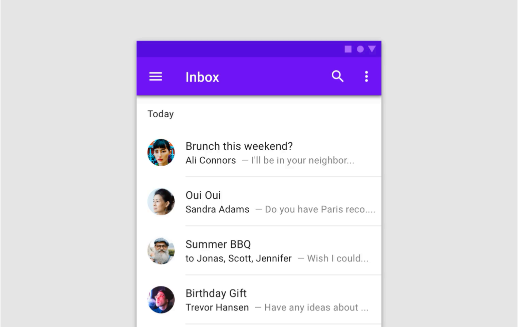
□ Eggs

To Bring:

Camping gear, lantern, smores, extra blankets, towel, warm socks, first aid kit

Clean desk before office tour!

Yuna tickets on sale



Inbox

Today

Brunch this weekend?
Ali Connors — I'll be in your neighbor...

Oui Oui
Sandra Adams — Do you have Paris reco...

Summer BBQ
to Jonas, Scott, Jennifer — Wish I could...

Birthday Gift
Trevor Hansen — Have any ideas about ...

Type of Animation

❑ TweenAnimation

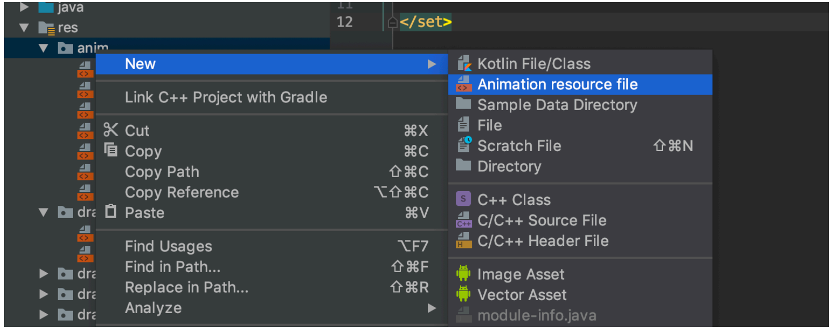
❑ FrameAnimation

❑ PropertyAnimation

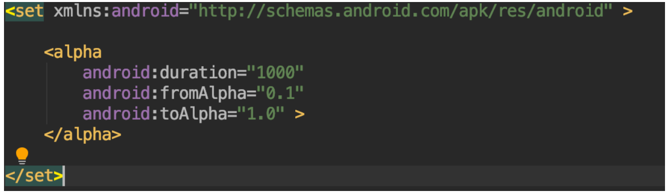
TweenAnimation

❑ AlphaAnimation

1.创建xml文档

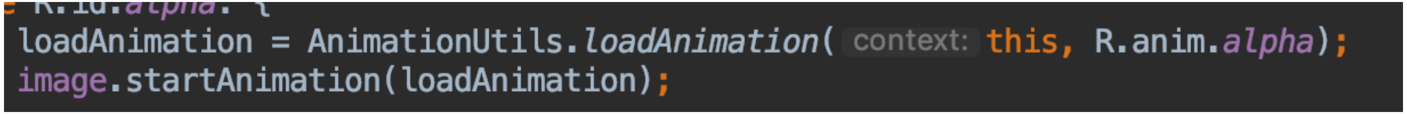


fromAlpha:初始alpha值
toAlpha:结束时alpha值
duration：延续时间



```
<set xmlns:android="http://schemas.android.com/apk/res/android" >

    <alpha
        android:duration="1000"
        android:fromAlpha="0.1"
        android:toAlpha="1.0" >
    </alpha>

</set>
```

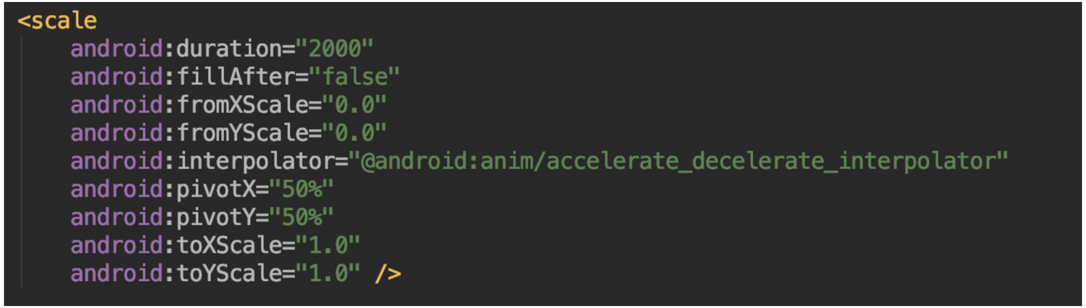
2.通过AnimationUtils.loadAnimation()加载



```
e R.id.alpha: {
loadAnimation = AnimationUtils.loadAnimation( context: this, R.anim.alpha);
image.startAnimation(loadAnimation);
```

TweenAnimation

❑ ScaleAnimation



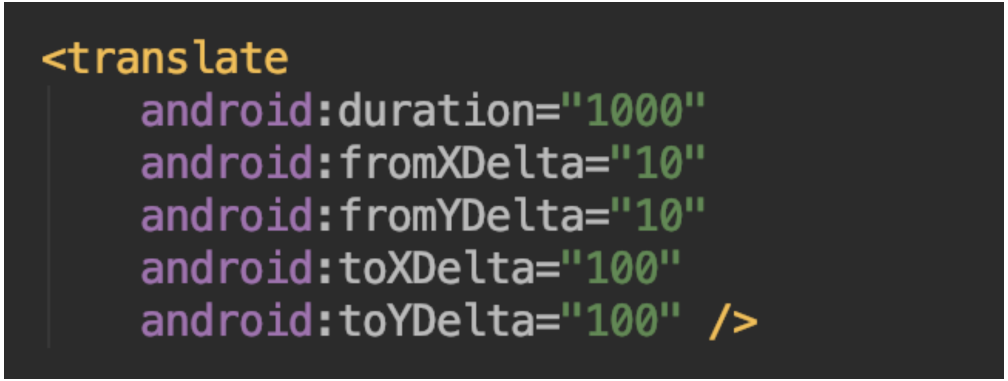
```
<scale
    android:duration="2000"
    android:fillAfter="false"
    android:fromXScale="0.0"
    android:fromYScale="0.0"
    android:interpolator="@android:anim/accelerate_decelerate_interpolator"
    android:pivotX="50%"
    android:pivotY="50%"
    android:toXScale="1.0"
    android:toYScale="1.0" />
```

fromXScale:起始x缩放比例
fromYScale:起始y缩放比例
toXScale:结束x缩放比例
toYScale:起始y缩放比例
pivotX:???
pivotY:???
fillAfter:???
fillbegin:???
Interpolator:???

- pivotX/Y:动画开始的起始坐标值，可以是整数值、百分数（或者小数）、百分数p 三种样式
- fillAfter/fillbegin:结束动画时的状态
- Interpolator:插值器，可以理解成动画变化的速率

TweenAnimation

❑ TranslateAnimation

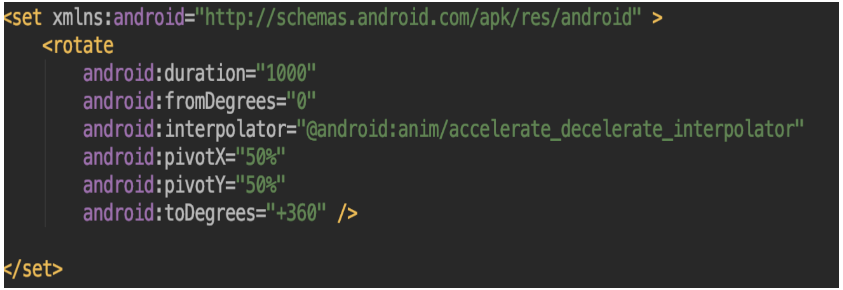


```
<translate
    android:duration="1000"
    android:fromXDelta="10"
    android:fromYDelta="10"
    android:toXDelta="100"
    android:toYDelta="100" />
```

fromXDelta: 起始x值
fromYDelta :起始y值
toXDelta:结束x值
toYDelta:结束y值

TweenAnimation

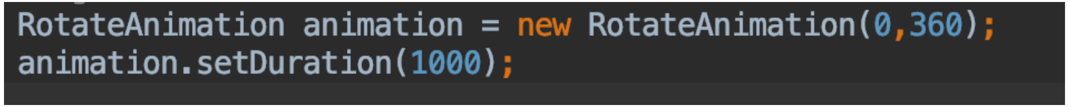
❑ RotateAnimation



```
<set xmlns:android="http://schemas.android.com/apk/res/android" >
    <rotate
        android:duration="1000"
        android:fromDegrees="0"
        android:interpolator="@android:anim/accelerate_decelerate_interpolator"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toDegrees="+360" />

</set>
```

fromDegree: 起始角度
toDegree:结束角度

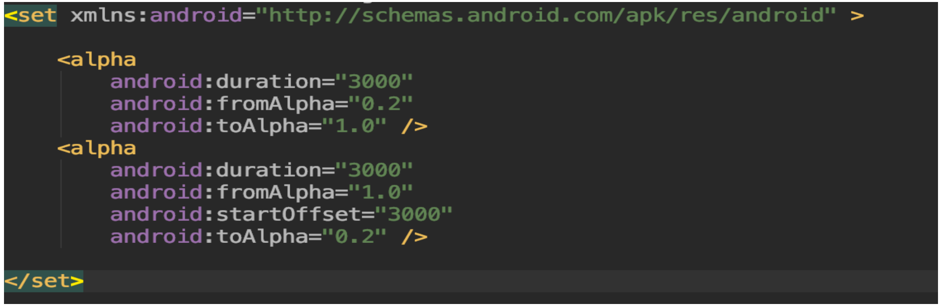
另一种实现方式：



```
RotateAnimation animation = new RotateAnimation(0,360);
animation.setDuration(1000);
```

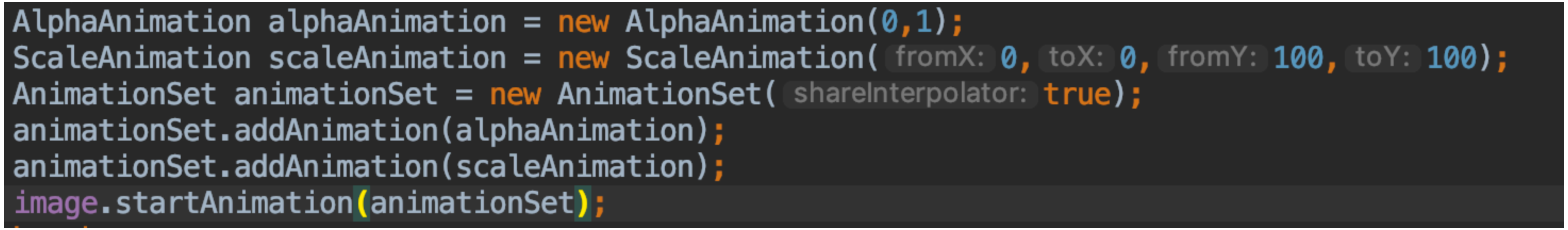
AnimationSet

1.利用xml文档进行自由组合



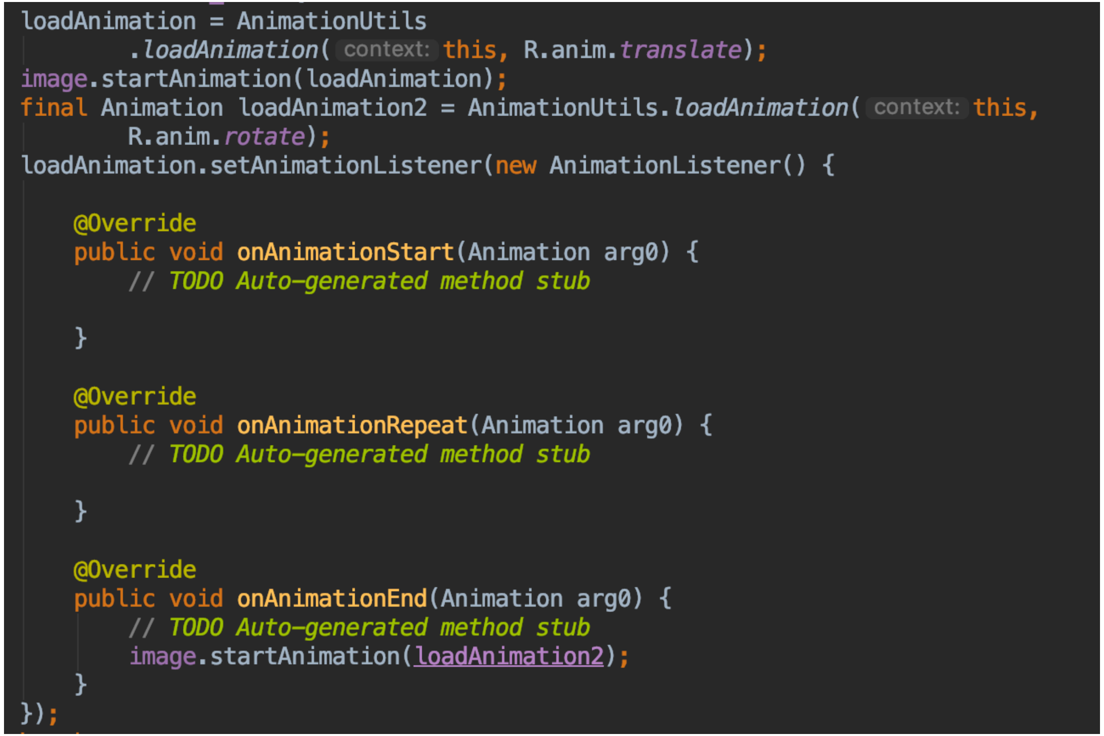
```
<set xmlns:android="http://schemas.android.com/apk/res/android" >

    <alpha
        android:duration="3000"
        android:fromAlpha="0.2"
        android:toAlpha="1.0" />
    <alpha
        android:duration="3000"
        android:fromAlpha="1.0"
        android:startOffset="3000"
        android:toAlpha="0.2" />

</set>
```

2.利用AnimationSet 进行自由组合



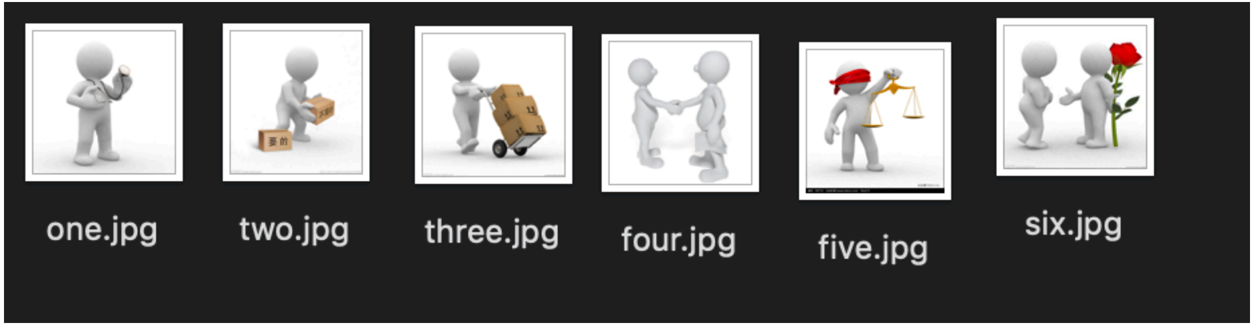
```
AlphaAnimation alphaAnimation = new AlphaAnimation(0,1);
ScaleAnimation scaleAnimation = new ScaleAnimation( fromX: 0, toX: 0, fromY: 100, toY: 100);
AnimationSet animationSet = new AnimationSet( shareInterpolator: true);
animationSet.addAnimation(alphaAnimation);
animationSet.addAnimation(scaleAnimation);
image.startAnimation(animationSet);
```

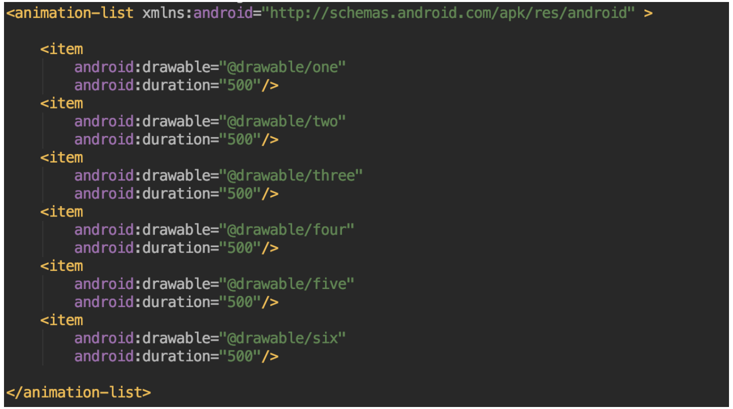
AnimationListener



```
loadAnimation = AnimationUtils
        .loadAnimation( context: this, R.anim.translate);
image.startAnimation(loadAnimation);
final Animation loadAnimation2 = AnimationUtils.loadAnimation( context: this,
        R.anim.rotate);
loadAnimation.setAnimationListener(new AnimationListener() {

    @Override
    public void onAnimationStart(Animation arg0) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onAnimationRepeat(Animation arg0) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onAnimationEnd(Animation arg0) {
        // TODO Auto-generated method stub
        image.startAnimation(loadAnimation2);
    }
});
```

FrameAnimation

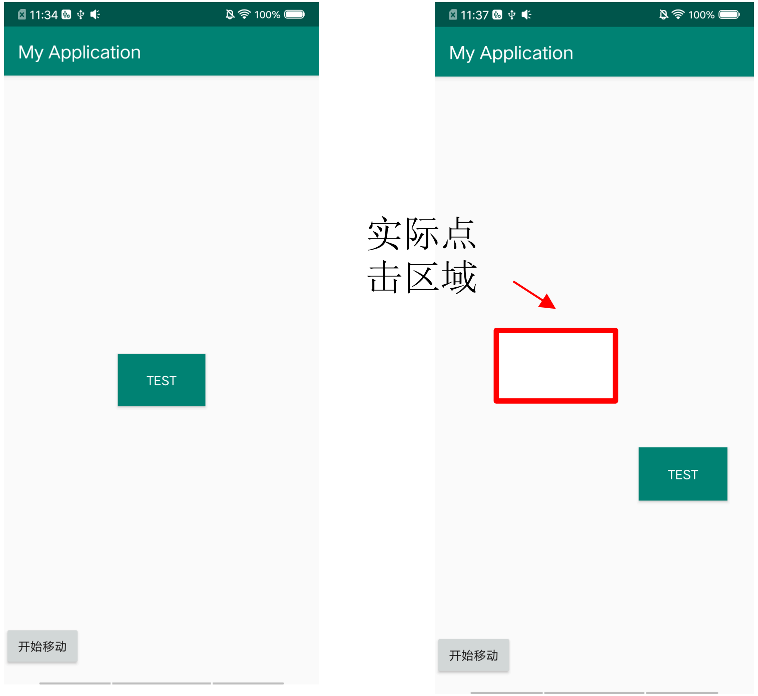




```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:drawable="@drawable/one"
        android:duration="500"/>
    <item
        android:drawable="@drawable/two"
        android:duration="500"/>
    <item
        android:drawable="@drawable/three"
        android:duration="500"/>
    <item
        android:drawable="@drawable/four"
        android:duration="500"/>
    <item
        android:drawable="@drawable/five"
        android:duration="500"/>
    <item
        android:drawable="@drawable/six"
        android:duration="500"/>

</animation-list>
```

PropertyAnimation

- ❏ ObjectAnimator
- ❏ ValueAnimator
- ❏ AnimatorSet

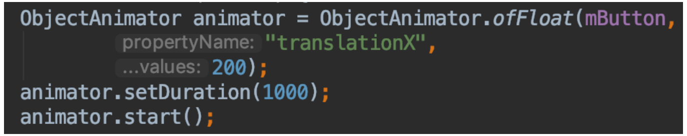


实际点击区域

PropertyAnimation

❑ ObjectAnimator

代码示例：



```
ObjectAnimator animator = ObjectAnimator.ofFloat(mButton,
        propertyName: "translationX",
        ...values: 200);
animator.setDuration(1000);
animator.start();
```

其他属性：

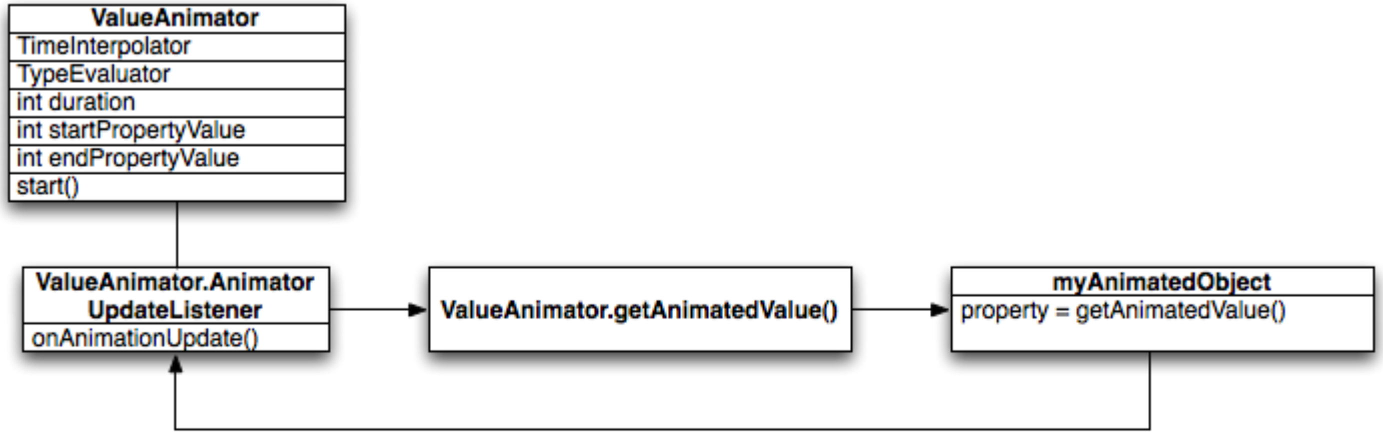
- translationX/translationY
- rotationX/rotationY
- scaleX/scaleY
- alpha
- ……

使用包装类：

```
private static class WrapperView {
    private View mTarget;

    public WrapperView(View mTarget) {
        this.mTarget = mTarget;
    }

    public int getHeight() {
        return mTarget.getLayoutParams().height;
    }

    public void setHeight(int height) {
        mTarget.getLayoutParams().height = height;
        mTarget.requestLayout();
    }
}
```

PropertyAnimation

- ValueAnimator



PropertyAnimation

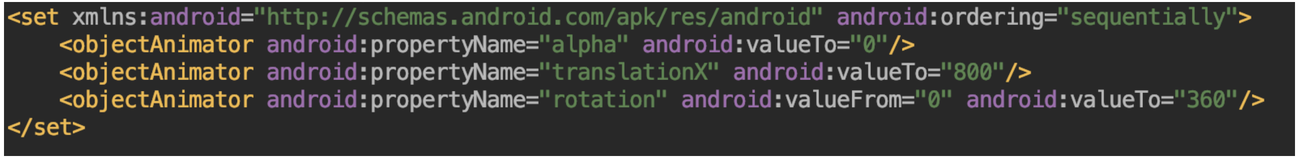
❑ ValueAnimator

```
ValueAnimator animation = ValueAnimator.ofFloat(0f, 100f);
animation.setDuration(1000);
animation.addUpdateListener(new
ValueAnimator.AnimatorUpdateListener() {
    @Override
    public void onAnimationUpdate(ValueAnimator updatedAnimation)
{
        float animatedValue =
(float)updatedAnimation.getAnimatedValue();
        textView.setTranslationX(animatedValue);
    }
});
animation.start();
```

PropertyAnimation

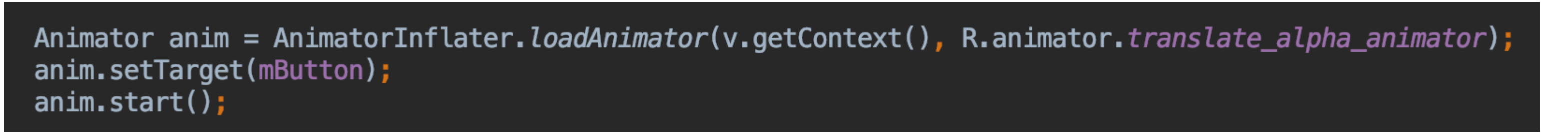
❑ AnimatorSet

1.创建translate_alpha_animator.xml文件



```
<set xmlns:android="http://schemas.android.com/apk/res/android" android:ordering="sequentially">
    <objectAnimator android:propertyName="alpha" android:valueTo="0"/>
    <objectAnimator android:propertyName="translationX" android:valueTo="800"/>
    <objectAnimator android:propertyName="rotation" android:valueFrom="0" android:valueTo="360"/>
</set>
```

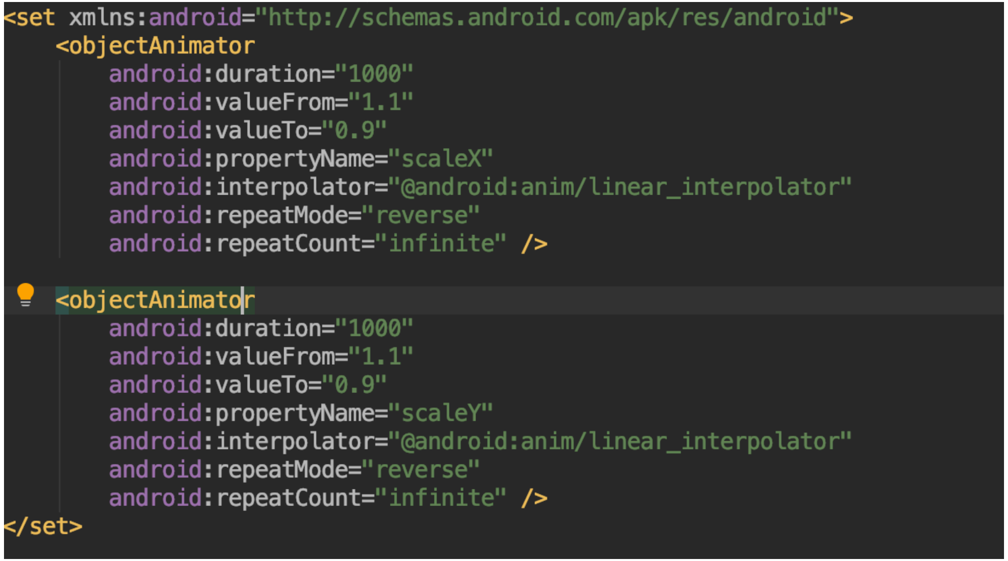
2.使用AnimationInflator加载



```
  Animator anim = AnimatorInflater.loadAnimator(v.getContext(), R.animator.translate_alpha_animator);
  anim.setTarget(mButton);
  anim.start();
```

PropertyAnimation

❑ AnimatorSet



```
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <objectAnimator
        android:duration="1000"
        android:valueFrom="1.1"
        android:valueTo="0.9"
        android:propertyName="scaleX"
        android:interpolator="@android:anim/linear_interpolator"
        android:repeatMode="reverse"
        android:repeatCount="infinite" />

    <objectAnimator
        android:duration="1000"
        android:valueFrom="1.1"
        android:valueTo="0.9"
        android:propertyName="scaleY"
        android:interpolator="@android:anim/linear_interpolator"
        android:repeatMode="reverse"
        android:repeatCount="infinite" />
</set>
```

- propertyName:”alpha””scaleX”…
- repeatCount:infinite
- repeatMode:reverse / restart
- valueFrome
- valueTo
- interpolator

PropertyAnimation

❏ AnimatorSet

```
AnimatorSet setAnimation = new AnimatorSet();
//示例1
setAnimation.play(translateAnimation).after(alphaAnimation).before(rotateAnimation);
setAnimation.play(rotateAnimation).before(scaleAnimation);

//示例2
setAnimation.playSequentially(alphaAnimation, translateAnimation, rotateAnimation,
scaleAnimation);

//示例3
setAnimation.playTogether(alphaAnimation, translateAnimation, rotateAnimation,
scaleAnimation);
```

Lottie

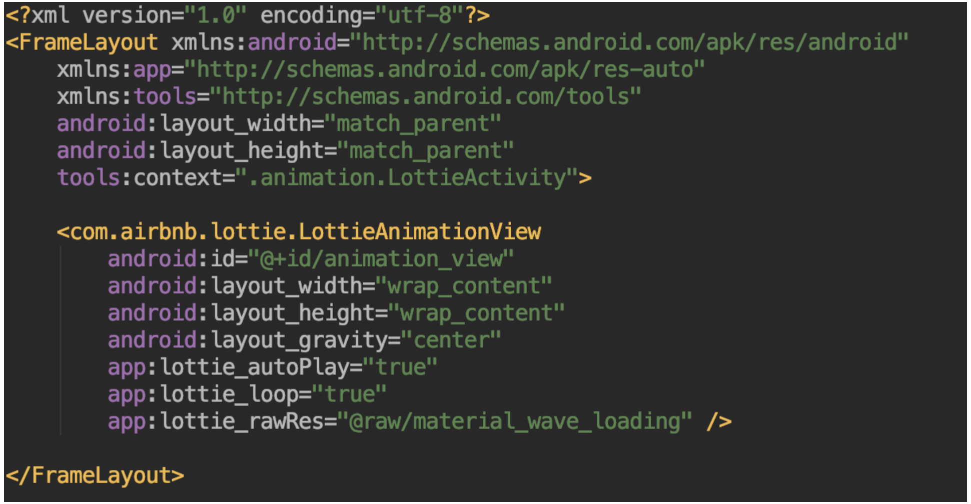
❑ build.gradle

```
dependencies {
        implementation 'com.airbnb.android:lottie:$lottieVersion’
 }
```

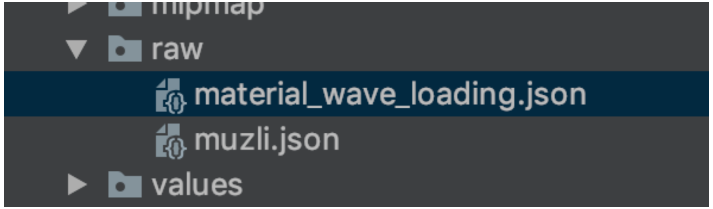
❑ LottieAnimationView

```
<com.airbnb.lottie.LottieAnimationView
android:id="@+id/animator_view"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
app:lottie_loop="true"
app:lottie_autoPlay="true" />
```

Demo-Lottie



```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".animation.LottieActivity">

    <com.airbnb.lottie.LottieAnimationView
        android:id="@+id/animation_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:lottie_autoPlay="true"
        app:lottie_loop="true"
        app:lottie_rawRes="@raw/material_wave_loading" />

</FrameLayout>
```



THANKS

ByteDance字节跳动